

Sample Final Exam

CS474 Spring 2023

Name:	UIN:
-------	------

Instructions: Fill in your name and UIN in the cover page and on every page. Read each question carefully. Please keep your answer within the provided boxes. This exam has a total of 6 questions (20 parts) worth 100 points.

Question 1. (20 points) The following Java code computes the maximum grade of a class:

```
class List<T> {
    T item;
    List<T> rest;
}

class Enrollment {
    Student s;
    int grade;
}

int maxGrade(List<Enrollment> roster) {
    if (roster == null)
        return 0;
    else
        return Math.max(roster.item.grade, maxGrade(roster.rest));
}
```

- 1.1. (5 points) Rewrite method `maxGrade` using the `Stream` methods `map` and `reduce`. As a reminder, the `map` method takes a function and applies it to each element in the target stream, while the `reduce` method takes a starting value and a binary operation and uses the operation to combine all the elements of the stream.
- 1.2. (5 points) Which of the stream methods you used returns another stream? Describe the contents of the stream it returns in your code.
- 1.3. (5 points) Suppose we wanted to make `Enrollment` generic in the type of the grade field, so that we can make `Enrollments` with grades of any type that implements the `Comparable` interface. How would you change the definition of `Enrollment` to do this?
- 1.4. (5 points) What are the advantages of the code you wrote over simply changing the type of `grade` to `Comparable`?

Question 2. (20 points) Suppose you had an interpreter for the programming language F# written in Java.

- 2.1. (5 points) Which language is the **host language** of the interpreter? Which language is the **guest language**?
- 2.2. (7 points) Suppose the interpreter's implementation of a `let` expression looked like this:

```
Value evaluate(LetExpression let, Environment e) {
    Value val = evaluate(let.def, e);
    return evaluate(let.body, e.bind(let.var, val));
}
```

What value would the interpreter return for an `Expression` representing the program `let x = 3 in (let x = 4 in 5) + x`?

- 2.3. (8 points) Suppose that instead of using the `Environment` data structure, our interpreter just kept a map from variables to their values, and implemented `let` by updating the map:

```
Map<Name, Value> valMap;
```

```
Value evaluate(LetExpression let) {
    Value val = evaluate(let.def);
    valMap.put(let.var, val);
    return evaluate(let.body);
}
```

What value would this modified interpreter return for an `Expression` representing the program `let x = 3 in (let x = 4 in 5) + x`? If the value is different, why? If not, why not?

Question 3. (19 points) Consider the following Java code.

```
for(int i = 0; i < 100; i++){
    a(i);
    for(int j = 0; j < 100; j++){
        b(i, j);
        for(int k = 0; k < 100; k++){
            c(i, j, k);
        }
    }
}
```

- 3.1. (4 points) Which of `a`, `b`, and `c` is the best candidate for optimization?
- 3.2. (5 points) Suppose we have two JVMs, one that does just-in-time compiling and one that does not. Which is likely to perform better in the first iteration of the outer loop? Why?

- 3.3. (5 points) Which is likely to perform better in the 100th iteration of the loop? Why?
- 3.4. (5 points) Describe a kind of program whose performance would **not** be improved by just-in-time compiling.

Question 4. (16 points) Suppose you were writing an interpreter for an object-oriented language. For each of the following elements, indicate whether you would prefer to store it in the representation of the **class** it belongs to or the representation of the **object** it belongs to (or both), and why.

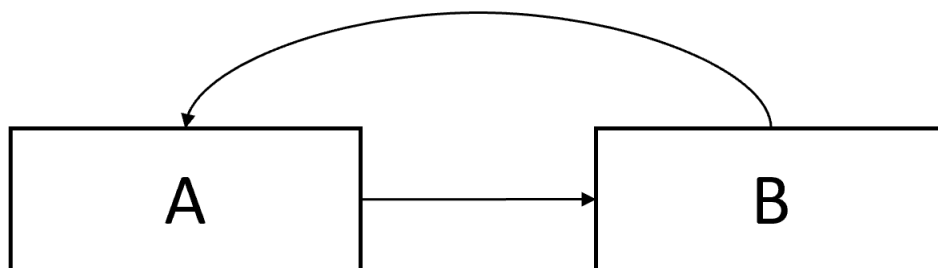
- 4.1. (4 points) field names
- 4.2. (4 points) field values
- 4.3. (4 points) method bodies
- 4.4. (4 points) superclass names

Question 5. (15 points) Consider the following JVM bytecode program.

```
0: iconst_2
1: iconst_1
2: iconst_3
3: invokestatic #1
```

- 5.1. (5 points) Draw the operand stack after executing lines 0, 1, and 2.
- 5.2. (5 points) Suppose #1 refers to a static method that takes two `int` arguments and returns their sum. Draw the operand stack after executing line 3.
- 5.3. (5 points) Suppose #1 referred to an instance method instead of a static method. What instruction would you use instead of `invokestatic`? Would that require any other changes to the listed bytecode?

Question 6. (10 points) Suppose a data structure with the following layout exists in the heap, with no pointers other than the ones shown:



- 6.1. (5 points) Would this data structure be considered garbage by a reference-counting approach?
- 6.2. (5 points) Would this data structure be considered garbage by a tracing approach?